

معرفی الگوریتم های فرهنگی و ارائه یک پیاده سازی جدید
با موضوع تکامل ذائقه غذایی افراد به هنگام حضور در جوامع اجتماعی

Introduce and Utilization of Cultural Algorithms
with Implementation of a new Algorithm

پروژه پایانی دوره کارشناسی

استاد راهنما : سرکارخانم دکتر مینا زلفی

علی فاضلی

دانشکده برق و مهندسی کامپیوتر دانشگاه تبریز

تابستان ۹۶





فهرست مطالب

| | |
|---|----|
| مقدمه : | ۵ |
| قسمت های مختلف الگوریتم فرهنگی: | ۵ |
| روش های تغییرات جمعیت در الگوریتم فرهنگی: | ۷ |
| ویژگی های اساسی الگوریتم فرهنگی: | ۸ |
| زمینه های مناسب برای بکارگیری الگوریتم فرهنگی: | ۹ |
| ساختار کلی پروژه: | ۱۰ |
| تاریخچه: | ۱۰ |
| پیشینه استفاده از مدل محاسباتی تکامل فرهنگی: | ۱۱ |
| مؤلفه های دانش الگوریتم فرهنگی: | ۱۳ |
| مؤلفه وضعی (Situational Component) : | ۱۳ |
| مؤلفه هنجاری (Normative Component) : | ۱۳ |
| مؤلفه دامنه اسمی دانش (Domain Knowledge) : | ۱۴ |
| مؤلفه های گزینش جمعیت در الگوریتم فرهنگی: | ۱۴ |
| تابع پذیرش (Acceptance Function) : | ۱۴ |
| بروز رسانی فضای اعتقادی (فرهنگ) : ۱- مؤلفه وضعی : | ۱۵ |
| تابع تاثیر (Influence Function) : | ۱۷ |
| صرفا استفاده از مؤلفه هنجاری : | ۱۷ |
| صرفا استفاده از مؤلفه وضعی برای تشخیص جهت حرکت: | ۱۷ |
| استفاده از مؤلفه موضعی برای تعیین جهت و مؤلفه هنجاری برای میزان حرکت: | ۱۸ |
| استفاده از مؤلفه هنجاری هم برای تعیین جهت و هم برای تعیین میزان گام: | ۱۸ |
| مراحل (روند اجرا) الگوریتم فرهنگی: | ۱۹ |
| پیاده سازی اولیه الگوریتم فرهنگی: | ۲۰ |
| نیازمندی های پیاده سازی: | ۲۰ |
| قسمت اول - تعریف مساله : | ۲۱ |
| قسمت دوم - تنظیمات مربوط به الگوریتم فرهنگی: | ۲۲ |
| قسمت سوم - آماده سازی : | ۲۳ |
| قسمت چهارم - بروزرسانی بهترین پاسخ یافت شده: | ۲۴ |

| | |
|--|---------|
| قسمت پنجم – آماده سازی فرهنگ: | ۲۵..... |
| علت وجود اسکریپت AdjustCulture.m در الگوریتم چیست؟ | ۲۶..... |
| اجرای الگوریتم فرهنگی در بستر نرم افزار متلب | ۲۸..... |
| پیاده سازی شبیه ساز تکامل دستورهای پخت غذا : | ۲۹..... |
| طرح مساله: | ۲۹..... |
| مراحل انجام الگوریتم فرهنگی بر روی دستورهای پخت: | ۳۱..... |
| نحوه اجرای الگوریتم: | ۳۲..... |
| نتیجه گیری و کارهای آینده: | ۳۴..... |
| منابع: | ۳۵..... |

مقدمه :

الگوریتم های ژنتیک از گونه الگوریتم های تکاملی می باشند که با استفاده از ایده نحوه تاثیر جهش ژنتیکی و بهبود آن بر روی اعضای جامعه و با نگاه ژنتیکی و طبیعی به موضوع تکامل انسان ها در طبیعت توانسته خیل عظیمی از مسائل انسان را به خوبی مدل کند. اما در الگوریتم های فرهنگی فرایند تکامل افراد جامعه بصورت دوگانه، هم از جهت تکامل ژنتیکی برخواسته از طبیعت انسان، و هم با استفاده از فرهنگ موجود در جامعه برخواسته از تمدن آن ها انجام می گیرد. منطقاً غربالگری و تکامل جمعیت با استفاده از دو فاکتور نتیجه بسیار مطلوب تری در بسیاری از مواقع بدست خواهد داد. که اینگونه الگوریتم ها را "الگوریتم فرهنگی" و یا "الگوریتم تکاملی فرهنگی" می نامند.

این الگوریتم ها موارد مشابه یا اصطلاحاً Counter Part های الگوریتم ژنتیک می باشند. همانطور که در الگوریتم ژنتیک شاهد حل مسائل با استفاده از تکامل زیستی هستیم در الگوریتم های فرهنگی نیز به دنبال استفاده از تکامل فرهنگی یک جامعه، فضای فرهنگی حاکم بر آن و تعامل و یا تقابل فرهنگ در برابر جامعه به منظور حل مسائل می باشیم. در اینجا منظور از جامعه تاحدود بسیار زیادی با مجموعه ژنوم ها در الگوریتم ژنتیک مشابهت دارد و این همان می تواند به عنوان هر پارامتری ایفای نقش کند. این مهم با استفاده از پیاده سازی و ساخت یک مدل انجام می گیرد که در نهایت موجب کمک به حل مسائل بهینه سازی خواهد شد.

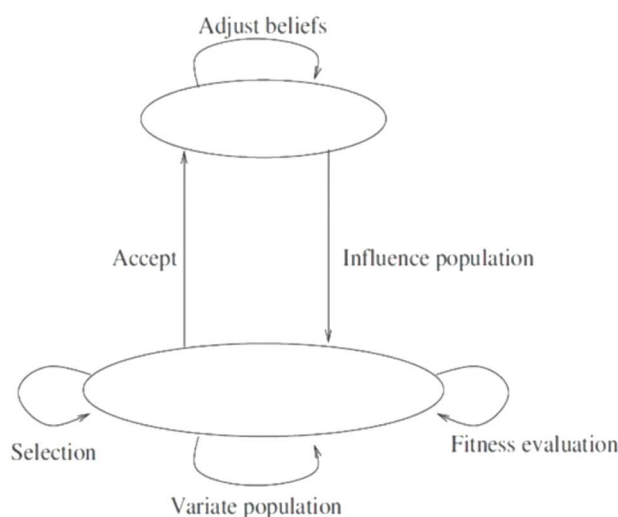
قسمت های مختلف الگوریتم فرهنگی:

در ابتدا جالب است که بحث را با چند سوال آغاز کنیم:

سوالی که در اینجا مطرح می شود این است که تکامل چگونه انجام می پذیرد؟ در یک اجتماع چه افرادی نقش بیشتری در تولید و تغییر فرهنگ دارند؟ الگوریتم فرهنگی پاسخ این سوال را در وجود "نخبگان جامعه" می داند. نخبه کسی است که در جامعه وزن و تاثیر بیشتری داشته باشد و یا به عبارتی بیشتر مورد توجه باشد. این گونه افراد "الگو" نیز نامیده می شوند. این فرد میتواند یک بازیگر یا یک سیاست مدار یا یک ورزشکار باشد که با استفاده از تمکن مالی خود و یا شهرت و یا قدرت نفوذ در اجتماع خود به اعمال سلیقه شخصی خود در رفتار عمومی جامعه بپردازد. طبیعتاً رفتار و جهت گیری های یک فرد مشهور و مورد توجه در جامعه بر روی جامعه تاثیر بسزایی داشته و آرام آرام رفتارش تبدیل به فرهنگ در جامعه می شود. این اتفاق با عنوان "تاثیر افراد بر روی فرهنگ" و "ایجاد مولفه فرهنگی" بررسی می شود.

اما در رابطه با "تاثیر فرهنگ بروی افراد" این توضیح وجود دارد که هنگامیکه رشته ای از افراد یک مولفه فرهنگی را در جامعه ایجاد می کنند. و یک فرهنگ بر جامعه غالب می شود، دسته ای از مردم به دلیل آنکه محل ارجاع رفتارهای اجتماعی شان فرهنگ می باشد، از این مولفه فرهنگی (فرهنگ) تاثیر پذیرفته و فرهنگ پیاده سازی می شود. و پس از آن افراد با اعمال سلیقه شخصی خود قسمت کوچکی از آن مولفه فرهنگی را با تغییرات کوچک شخصی سازی کرده و از آن پیروی می کنند.

فرض کنید یک فرد در یک جامعه یک مولفه فرهنگی جدید (مانند استفاده از کتب امانی در وسایل نقلیه عمومی) را می پذیرد. اما طبیعتاً هر فرد مقداری از سلیقه شخصی خود را در فرهنگ غالب جامعه دخیل می کند. بطور مثال این فرد یک تکه سفید برداشته، بخش های جالب کتاب عمومی را در آن یادداشت کرده و دوباره لای کتاب می گذارد. این مولفه فرهنگی شخصی سازی شده (در موارد مثبت می گوییم بهبود یافته) در جامعه پذیرفته می شود و هر فرد دیگری که این مولفه فرهنگی را می پذیرد (استفاده از کتب عمومی در وسایل نقلیه) عمل نکته برداری نکات جالب کتاب را نیز انجام می دهد. و این گونه است که بهبود یک مولفه فرهنگی انجام می پذیرد. در کل اساس کار الگوریتم های فرهنگی همین نکته می باشد.



تصویر ۱ ساختار کلی یک الگوریتم فرهنگی

در شکل بالا قسمت بالایی شکل نشان دهنده فرهنگ و قسمت پایین نشان دهنده جمعیت می باشد. ما یک بده بستان بین فرهنگ و جمعیت داریم. تاثیری که جمعیت بر فرهنگ دارد اصطلاحاً اثر acceptance یا مقبولیت می باشد (همرنگ جماعت شو). و بطور متقابل تاثیری که فرهنگ بر روی جمعیت مورد مطالعه دارد را تاثیر یا influence گویند.

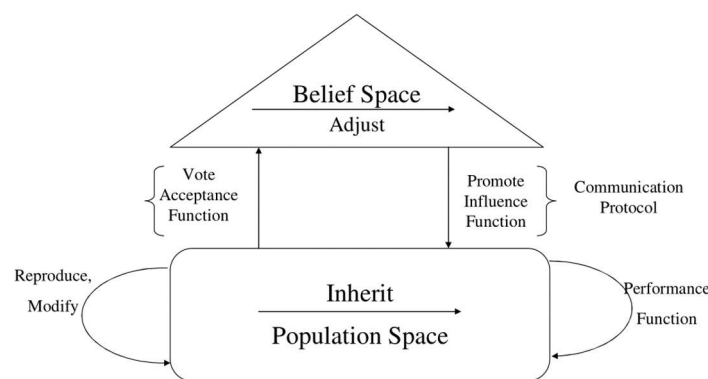
بطور مثال، افرادی که وزن بیشتری از لحاظ ارزشی در جامعه دارند (به عبارتی نخبه هستند) ، نرخ مقبولیت بیشتری نسبت به دیگر اعضای جامعه دارند.

تغییراتی که در فرهنگ انجام می پذیرد و تاثیراتی که فرهنگ بر روی فرهنگ می گذارد را Adjust Belief می گویند. گاهی فرهنگ جدیدی بوجود می آید که فرهنگ های دیگر را تحت تاثیر قرار می دهد.

روش های تغییرات جمعیت در الگوریتم فرهنگی:

از جمله تغییراتی که در جمعیت رخ میدهد می توان Selection ، Variate Population و Fitness Evaluation را نام برد که در ادامه به هر کدام از آنان اشاره خواهیم کرد:

- Selection یا انتخاب : فرآیند وزن دهی به عناصر جامعه را گویند. اتخاذ فرایندی که طی آن عناصر مهم جامعه از دیگر عناصر متمایز شوند (و متعاقبا تاثیر بیشتری بر روی فرهنگ داشته باشند) را انتخاب گویند.
- Fitness Evaluation: به معنای ارزیابی جمعیت بهتر می باشد.
- و در آخر قسمت Variate Population که به معنای تغییر اعضای جمعیت می باشد که در آن شیوه هایی همچون mutation و یا crossover انجام پذیرد. این Variation ها در الگوریتم ژنتیک کلاسیک نیز وجود دارد. اما دخالت دادن فرهنگ و مقادیر بهینه در فرایند اصلاح رویکردی است که در الگوریتم های فرهنگی بدان پرداخته می شود.

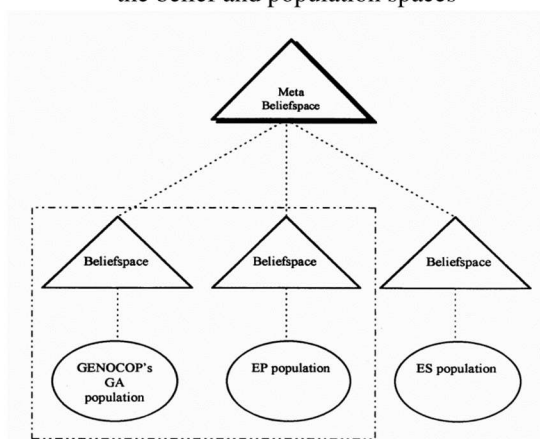


تصویر ۲ اشکال مختلف ارتباطات بین اعضای الگوریتم فرهنگی

ویژگی های اساسی الگوریتم فرهنگی:

- تاثیر پذیری و ارث بری دوگانه، از طریق جمعیت و سطوح دانش (در این الگوریتم بالاخص فرهنگ)
- دانش یا همان فرهنگ به عنوان چراغ راه مسیردهی به اعضای جمعیت به منظور بهبود روند جست و جو می باشد
- قابلیت پشتیبانی از طراحی روال جست و جوی سلسله مراتبی براساس جمعیت و یا فضای اعتقادی(فرهنگ)

Can support the emergence of hierarchical structures in both the belief and population spaces

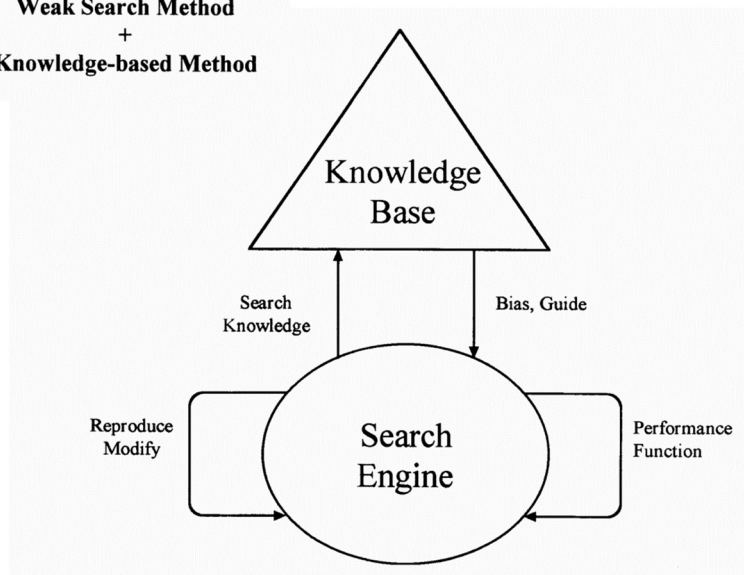


تصویر ۳ نحوه ارتباطات سلسله مراتبی بین اعضای الگوریتم فرهنگی

- جداسازی دانش برگزیده (فرهنگ) از دانش فردی (فرهنگ هریک از اعضای جمعیت)
- پشتیبانی از انطباق خودکار مولفه فرهنگی در سطوح مختلف
- امکان انجام تکامل فرهنگی با نرخ ها و سطوح مختلف
- پشتیبانی از رویکرد های ترکیبی (hybrid) برای حل مسائل

Hybrid System:

Weak Search Method
+
Knowledge-based Method



تصویر ۴ پشتیبانی از رویکرد های ترکیبی برای حل مسائل

زمینه های مناسب برای بکارگیری الگوریتم فرهنگی:

ویژگی های ذاتی الگوریتم فرهنگی (که در بالا به آن اشاره شد) باعث می شود که این الگوریتم پتانسیل بالایی برای حل مسائل زیر داشته باشد:

- مسائلی که در آن ها مقادیر domain knowledge قابل توجه و زیاد باشد. بطور مثال در مسائل بهینه سازی مطلق (اجباری) (Constrained optimization problems)
- مسائل پیچیده مربوط به سیستم هاییکه که عمل انطباق در سطوح و نرخ های مختلف در جمعیت و یا فضای اعتقادی صورت می گیرد.
- دانش در دسترسی دارای حالات و اشکال مختلف باشد و با معیارها و رویکردهای متفاوت مورد تصمیم گیری واقع شود
- سیستم های ترکیبی (hybrid) ای که نیاز به ترکیبی از جست و جوی صرف و چهارچوب های برپایه دانش دارند.
- حل مسائلی که نیازمند چندین گروه مختلف جمعیت و فضای اعتقادی، و متعاقبا ارتباط بین آن ها دارد.
- مسائل دارای ساختار سلسله مراتبی که ساختار جمعیت و یا روند تاثیرپذیری یا تاثیرگذاری آنها بروی فرهنگ بصورت سلسله مراتبی مدل می شوند.

ساختار کلی پروژه:

در ابتدا مقدمات و مفاهیم مورد نیاز برای درک الگوریتم فرهنگی مطرح شد، سپس به تاریخچه مختصری از نحوه پیدایش این نوع الگوریتم پرداخته میشود. سپس الگوریتم اولیه در نرم افزار متلب پیاده سازی و شرح داده می شود تا از تمامی جوانب و راه حل های موجود برای مواجهه با مساله با استفاده از الگوریتم فرهنگی با خبر شویم. در آخر مدل پیشنهادی اینجانب که استفاده کاربردی از الگوریتم های فرهنگی به منظور کاربردی سازی یک مفهوم نظری می باشد را شاهد هستیم.

تاریخچه:

در سال ۱۸۸۱ فردی بنام ادوارد ب. تیلور اولین دانشمند و جامعه شناسی بود که از لفظ "فرهنگ" استفاده نمود. او در کتابش بنام "فرهنگ اولیه" در همان سال فرهنگ را بصورت زیر تعریف کرد:

"مجموعه ای مرکب از دانش، اعتقادات، هنر، اخلاقیات، رسوم و دیگر قابلیت ها و عادات های بشر به عنوان جزئی از جامعه متمدن."

در سال ۱۹۵۷ جورج موردی کتابی منتشر نمود که ۵۶۵ فرهنگ مختلف بر اساس شخصیت ۳۰ نفر، مورد بررسی قرار گرفته بود.

در سال ۱۹۶۰ علم بررسی عادات و نحوه زندگی (اصطلاحاً Cultural Ecology) بصورت جدی به شکل تعاملات بین نظام فرهنگی جامعه و محیط بررسی شدند.

برخی از تعاریف محققان در این زمینه نیز در مطالعات بسیار حائز اهمیت است. که از این جمله تعاریف میتوان به تعریف گریتز در سال ۱۹۷۳ و تعریف دورهام در سال ۱۹۹۰ اشاره کرد:

"فرهنگ معنای مطلق تفسیری است که انسان ها از تجربیات و رفتار خود دارند." گریتز ۱۹۷۳

"فرهنگ به معنای به اشتراک گذاشتن پدیده های مشترک اتفاق افتاده در برابر انسان ها می باشد" دورهام ۱۹۹۰

سوال هائیکه در طراحی یک الگوریتم فرهنگی پیش می آید این است که :

- چگونه برای یک عضو مفروض جامعه عملیات مقبول شدن (acceptance) انجام می پذیرد. و چگونه در تغییر فرهنگ نقش دارد.

- اعضای جمعیت را با هدف تاثیرگذاری در فرهنگ چگونه انتخاب کنیم؟ سیاست گذاری ما برای انتخاب افراد در جمعیت چگونه باید باشد؟
- چگونه تغییرات فرهنگی را بر روی جمعیت اعمال کنیم؟

در الگوریتم فرهنگی قسمت فرهنگ یا Culture نام گذاری کلی ای است که بصورت دقیق تر و استانداردسازی شده در الگوریتم های تکاملی با نام فضای اعتقادی یا Belief Space خوانده می شود. که ویژگی های توصیفی و توصیف کنندگی آن بیشتر مد نظر می باشد. لذا ما یک فضای اعتقادی داریم که حاوی یک سری مولفه های دانش (Knowledge Components) می باشد.

پیشینه استفاده از مدل محاسباتی تکامل فرهنگی:

پیشینه مدل سازی محاسباتی بر اساس تکامل فرهنگی به سال ۱۹۸۰ بر می گردد که کاوالی اسفروزا و فلدمن (Cavalli-Sforza & Feldman, ۱۹۸۱) اولین محاسبات کمی وابسته به یک مدل را با استفاده یک الگوریتم فرهنگی ارائه دادند. که این امر با استناد بر اسناد ایشان و با استفاده از تحقیقات بسیار گسترده در زمینه اجتماع و روانشناسی (و به خصوص تکامل در زمینه تکامل زبان) صورت گرفته بوده است.

Cavalli-Sforza و Feldman مدل هایی از نحوه برخورد و تعامل فرهنگی انسان ها استخراج و تولید کردند که نه تنها تکامل با استفاده از انتقال هنجارها بوسیله والدین بیولوژیکی افراد (یادگیری اجتماعی عمودی) صورت می گرفت، بلکه این امر از هم سطح های یک عامل نیز صورت می گرفته است (یادگیری اجتماعی افقی) و حتی با استفاده از اعضای نامربوط از عوامل هم سطح والدین عامل و یا نیاکان آنها (که اصطلاحا یادگیری اجتماعی اریب نامیده می شود).

آنها مدل های جهش فرهنگی را همانند جهش (mutation) ژنتیکی که از قبل موجود بود پیاده سازی کردند. درحالیکه رفتارهای اجتماعی روز دهه ۸۰ میلادی بیشتر با الگوی تصادفی مدل می شدند، اسفروزا و فلدمن غربال عوامل را با استفاده از انتخاب طبیعت (هر عاملی که دارای قدرت و میزان سازگاری با محیط بیشتری است باقی مانده و پیروز می شود) پایه گذاری کردند. بدین نحو که هنجار اجتماعی بر اساس شانس بقایی که در میان جامعه دارد می تواند تغییر کند.

در سال ۱۹۸۸ بوید و ریچاردسون روانشناسی واقع گرایانه را وارد الگوریتم تکاملی فرهنگی کردند. این امر با استناد بر اینکه عوامل مورد بررسی در الگوریتم (در اینجا مردم) دارای این توانایی هستند که با قائل شدن امتیازی برای خود از رفتار و هنجار افراد با امتیاز بالاتر از خود (شان اجتماعی بالاتر، همانند الگوها و افراد شناخته شده جامعه) تقلید کنند.

این تقلید با استفاده از معیارهای مختلفی صورت می گیرد. روش اول که در بالا به آن اشاره شد روش معیار غیر مسیقیم (Prestige bias) می باشد. روش معیارگزینی بعدی روشی است که در آن عوامل هنجارهای اجتماعی را بر اساس محبوبیت عوامل برای خود بر می گزینند (این روش، مستقل از فراوانی و یا انطباق نیز نامیده می شود). روش معیارگزینی بعدی روشی است که انتخاب هنجار با استفاده از مشخصات و کاراکتر ذاتی عوامل صورت می گیرد. بدین صورت که که برخی ویژگی های خاص برخی عوامل مهم تعدا برجسته شده و عوامل جامعه مجبور به تقلید از آن می شوند. از این دسته از معیارگزینی میتوان به قوانین اساسی جامعه و خط قرمزهای آن اشاره کرد.

بوید و ریچاردسون همچنین مدلی ارائه کردند که در آن دو مکانیزم جداگانه برای انتقال و پذیرفتن هنجارها برای عوامل جامعه ارائه شد:

۱. جهش فرهنگی تصادفی که بر اساس مدل قبلی اسفروزا و فلدمن در سال ۱۹۸۰ ارائه شده بود پایه گذاری شد.

۲. دگرگونی تعلیمی (guided variation): که در این مدل افراد جامعه اطلاعات لازمه برای رسیدن به اهداف غایی و یا تمایلات خود را خودشان بدست می آورند.

در مدل ارائه شده در این مقاله، تمامی نقطه نظر های مختلف این اصول پیاده سازی شده است: Variation، Selection و Replication.

برای ساده سازی هدف، دگرگونی (Variation) فقط مقادیر ورودی خود را از جهش فرهنگی بصورت تصادفی دریافت می کند. و فرایند انتخاب (Selection) براساس فرایند انطباق (conformity bias) صورت می گیرد. به عبارت ساده تر انتقال یک هنجار اجتماعی به کل اعضای جامعه در ابتدا باید در یک گروه اجتماعی خاص امتحان خود را پس بدهد، سپس این انتقال هنجار انجام گیرد. جزئیات بیشتر در قسمت های بعدی ارائه می شوند.

مولفه های دانش الگوریتم فرهنگی:

مولفه وضعی (Situational Component):

از مهم ترین مولفه های الگوریتم فرهنگی می توان به "مولفه وضعی" یا Situational Component اشاره کرد که آمار ویژگی های شاخص در فرهنگ را مشخص می کند. بطور مثال باهوش ترین فرد درون جمعیت یا مشهورترین عنصر جامعه و امثال این گونه ویژگی ها.

در الگوریتم های تکاملی پیشین همواره مقادیر اکسترمم مطلق (حداقل ها و حداکثرها) متمایز شده و از دیگر اعضای جمعیت بگونه ای (باتوجه به نحوه پیاده سازی) جدا می شدند. اما در الگوریتم های فرهنگی علاوه بر نگهداری این مقادیر مقادیر بهینه در هرمرحله نیز نگهداری شده و از آن ها به منظور بهبود کارکرد الگوریتم استفاده می شود. یا به عبارت دیگر اگر به الگوریتم ژنتیک کلاسیک یک فضای فرهنگی برای تصمیم گیری ها و تغییرات اضافه کنیم الگوریتم فرهنگی را خواهیم داشت. بطور مثال در الگوریتم PSO به جای اینکه global best و personal best داشته باشیم، میتوان نقطه ای را به عنوان هدف در نظر گرفت که فرهنگ پیشنهاد می کند.

مولفه هنجاری (Normative Component):

بطورمثال این مولفه "بازه تحمل رانندگان در زمان حرکت اتومبیل جلویی خود" را مدل می کند. بطور مثال به ازای سرعت 15 km/h بایستی یک خودرو به اندازه طول خودرو خود از خودرو مقابل خود فاصله داشته باشد. و بطور مثال در سرعت 60 km/h بایستی این فاصله به اندازه چهار برابر طول خودرو باشد. در صورتیکه جوامعی وجود دارند که رعایت این قانون نانوشته به ضرر راننده و سوء استفاده دیگر رانندگان منجر می شود. حال چه خوب یا بد این هنجار برای یک فرهنگ غالب جامعه می باشد. و طبیعتا مقادیر کمی یا کیفی این هنجار می تواند به عنوان یک مولفه هنجاری در الگوریتم فرهنگی مورد استفاده قرار گیرد.

مولفه دامنه اسمی دانش (Domain Knowledge):

که بسیار شبیه به مولفه وضعی می باشد. بدین صورت تعریف می شود که در هر تکرار و هر نسل از جمعیت مقادیر بهینه گزارش شده و در این مولفه نقش دارد. به عبارت دیگر هر هنگام که مقادیر اکسترمم در هر نسل را ذخیره کرده و از آن ها برای بهبود استفاده کنیم می گوییم از مولفه دامنه اسمی دانش استفاده کرده ایم.

مولفه های گزینش جمعیت در الگوریتم فرهنگی:

تابع پذیرش (Acceptance Function):

عبارت است از انتخاب یک عضو از جمعیت به منظور تاثیر گذاری بر روی فرهنگ و انتخاب اینکه چه افرادی به چه نحوی می توانند بر روی فرهنگ تاثیر بگذارند. تعداد افراد تاثیر گذار بر روی فرهنگ به دو روش ایستا و پویا می تواند مدیریت شود. به شکل ایستا همانطور که از نام بر می آید اینگونه است که تعداد افرادی مشخص از تمام جمعیت به عنوان افراد بهینه متمایز شده و از آنها برای تاثیر گذاری در فضای فرهنگی مورد استفاده قرار می گیرد. و در حالت پویا تعداد افراد موثر ثابت نیست و با استفاده از الگوریتم هایی کمیت و کیفیت این تاثیر از اعضای جمعیت متغیر خواهد بود.

البته مولفه های دیگری بصورت ابتکاری ابداع شده اند که تعداد آنان بسیار بوده و از حوصله این بحث خارج است.

بروز رسانی فضای اعتقادی (فرهنگ) :

۱- مولفه وضعی :

همانطور که پیشتر هم اشاره شد مولفه وضعی یک رکود از بهترین پاسخ های فعلی تولید شده در مساله می باشد. پس این مولفه بهترین پاسخ تمام نسل ها و یا نسل جاری می باشد.

۲- مولفه هنجاری :

$$\vec{x} = (x_1, x_2, x_3, x_4, \dots, x_n)$$

$$N = \{X_1, X_2, X_3, X_4, \dots, X_n\}$$

$$X_i = (I_i, L_i, U_i)$$

$$I_i = (x_i^{min}, x_i^{max})$$

که در اینجا مولفه L_i و I_i همان مقدار حداقل و حداکثری برای متغیر x در حوزه i می باشد.

$$x_j^{min}(t) \cdot x_l(t) \rightarrow x_j^{min}(t+1)$$

اگر مقدار حد قبلی پاسخ را از قبل داشته باشیم و مقدار یک پاسخ جدید پذیرفته شده را نیز بپذیریم. می توان به مقدار پاسخ سیستم به مولفه j در مومنتوم زمانی بعدی نیز دست پیدا کنیم.

به عبارت دیگر خواهیم داشت:

$$x_j^{min}(t+1) = \begin{cases} x_{lj}(t) & x_{lj}(t) \ll x_j^{min}(t) \\ x_j^{min}(t) & otherwise \end{cases}$$

تنظیم فضای اعتقادی:

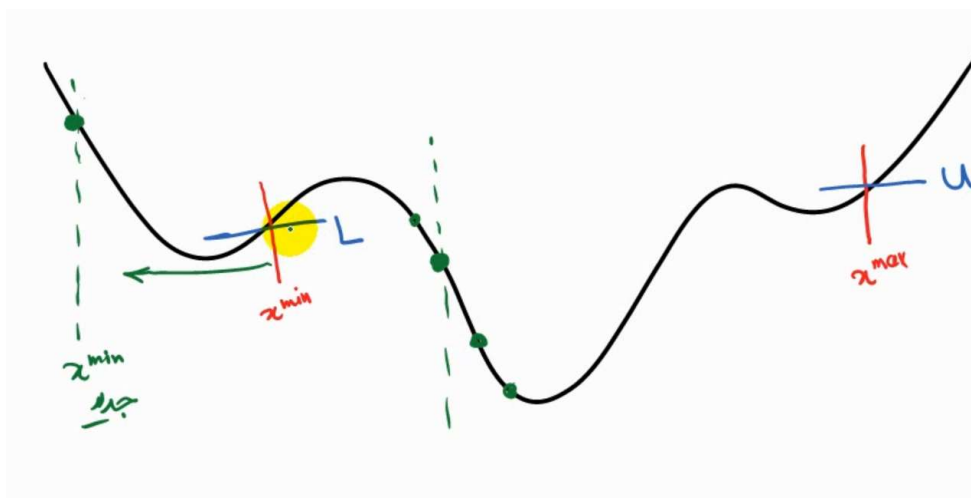
ضوابط مربوط به حد بالا و پایین جست و جو و حد بالا و پایین مطلق به شکل زیر محاسبه می گردند:

$$x_j^{min}(t+1) = \begin{cases} x_{l_j}(t) & x_{l_j}(t) \ll x_j^{min}(t) \text{ Or } f(x_{l_j}(t)) \ll L_j(t) \\ x_j^{min}(t) & \text{otherwise} \end{cases}$$

$$L_j(t+1) = \begin{cases} f(x_{l_j}(t)) & x_{l_j}(t) \ll x_j^{min}(t) \text{ Or } f(x_{l_j}(t)) \ll L_j(t) \\ L_j(t) & \text{otherwise} \end{cases}$$

$$x_j^{max}(t+1) = \begin{cases} x_{l_j}(t) & x_{l_j}(t) \gg x_j^{min}(t) \text{ Or } f(x_{l_j}(t)) < L_j(t) \\ x_j^{max}(t) & \text{otherwise} \end{cases}$$

$$U_j(t+1) = \begin{cases} f(x_{l_j}(t)) & x_{l_j}(t) \gg x_j^{min}(t) \text{ Or } f(x_{l_j}(t)) \ll L_j(t) \\ u_j(t) & \text{otherwise} \end{cases}$$



تصویر ۵ شماتیک کلی از نحوه جست و جو و غربالگری در الگوریتم فرهنگی

$$B(t) = \{S(t).N(t)\}$$

مولفه های تشکیل دهنده فضای اعتقادی عبارت است از :

$S(t)$: *Situational Component* بهترین پاسخ یافته شده تا کنون

$N(t)$: *Normative Component* حد بالا و حد پایین بازه بررسی

با استفاده از $B(t)$ میتوان یک فضای اعتقادی یا فرهنگ را توصیف کرد.

تابع تاثیر (INFLUENCE FUNCTION) :

تابع تاثیر روش های تغییر پاسخ ها و یا به عبارتی روش های تاثیر فرهنگ را روی جامعه مدل می کند. از این روش ها (که اصطلاحاً روش های تغییر پاسخ ها نام دارند) میتوان به موارد زیر اشاره کرد:

صرفاً استفاده از مولفه هنجاری :

می خواهیم مولفه زام پاسخی را تغییر بدهیم. و مقدار جدید آن را با پریم نشان می دهیم. (مقدار قدیمی پریم ندارد)

$$x'_{ij}(t) = x_{ij}(t) + \text{size}(\mathcal{I}_j(t))N_{ij}(0, 1)$$

where

$$\text{size}(\mathcal{I}_j(t)) = x_{\max,j}(t) - x_{\min,j}(t)$$

تصویر ۶ استفاده از مولفه هنجاری برای محاسبه تابع تاثیر

جواب جدید برابر است با جواب قدیم به علاوه طول بازه (گام) ضربدر یک عدد تصادفی نرمال استاندارد.

$$x'_{ij}(t) = x_{ij}(t) + \sigma_j(t) * N(0, 1)$$

$$\sigma_j(t) = \alpha[x_j^{\max}(t) - x_j^{\min}(t)]$$

صرفاً استفاده از مولفه وضعی برای تشخیص جهت حرکت:

اگر پاسخ بدست آمده کمتر از پاسخ قبلی ما باشد، بایستی پاسخ را با مقدار مثبتی جمع کرده تا به مقدار مطلوب برسیم. به همین ترتیب اگر پاسخ بدست آمده بیشتر از پاسخ قبلی ما باشد، بایستی حاصل مقدار جدید از عدد معین مثبتی کم شود تا به نتیجه دلخواه دست پیدا کنیم.

$$\sigma_{ij} > 0 : \text{Strategy Parameter}$$

$$x'_{ij}(t) = \begin{cases} x_{ij}(t) + \sigma_{ij}|N(0,1)| & x_{ij}(t) < y_j(t) \\ x_{ij}(t) - \sigma_{ij}|N(0,1)| & x_{ij}(t) > y_j(t) \\ x_{ij}(t) + \sigma_{ij}N(0,1) & otherwise \end{cases}$$

در حالت اول گام مثبت، در حالت دوم گام منفی، و در حالت سوم گام مثبت و منفی با احتمال یکسان میباشد.

استفاده از مولفه موضعی برای تعیین جهت و مولفه هنجاری برای میزان حرکت:

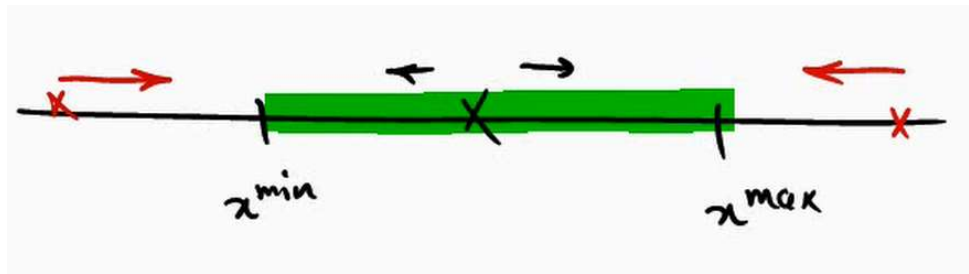
همان استفاده از مولفه موضعی است، با این تفاوت که برای مقدار دهی سیگما، برخلاف حالت دوم که مقداری ثابت در نظر گرفته می شد، در این روش از یک ضریب و مقدار متغیر برای مقداردهی ضریب تغییر در پاسخ ها استفاده می کنیم.

$$\sigma_{ij}(t) = \alpha [x_j^{max}(t) - x_j^{min}(t)]$$

و برای محاسبه پاسخ از همان روابط روش دوم استفاده می کنیم و تنها تغییر انجام در نحوه محاسبه مقدار سیگما می باشد.

استفاده از مولفه هنجاری هم برای تعیین جهت و هم برای تعیین میزان گام:

فرض کنید بازه هنجار مساله مورد بررسی ما بازه سبز رنگ نشان داده شده در شکل زیر باشد. اگر ما در خارج از این بازه قرار داشته باشیم بایستی با مقادیر جابه جا کننده مناسب به بازه هنجار مورد بررسی منتقل شویم. بطوریکه که اگر پاسخ مورد بررسی کوچکتر و با کمتر از بازه هنجار باشد بایستی به سمت راست حرکت کرده و به همان ترتیب تصمیم مناسب را برای مقادیر بزرگتر از پاسخ های هنجار اتخاذ کنیم.



تصویر ۷ استفاده از مولفه هنجاری هم برای تعیین جهت و هم برای تعیین میزان گام

اما زمانی که در بازه هنجار قرار داریم می توانیم به هرجهتی (چه به سمت پاسخ های بزرگتر و چه به سمت پاسخ های کوچکتر) حرکت کنیم. که این امکان با استفاده از یک ضریب به نام ضریب بتا برای الگوریتم فراهم می شود.

$$x_{ij}(t) = \begin{cases} x_{ij}(t) + |\text{size}(\mathcal{I}_j(t))N_{ij}(0,1)| & \text{if } x_{ij}(t) < x_{min,j}(t) \\ x_{ij}(t) - |\text{size}(\mathcal{I}_j(t))N_{ij}(0,1)| & \text{if } x_{ij}(t) > x_{max,j}(t) \\ x_{ij}(t) + \beta \text{size}(\mathcal{I}_j(t))N_{ij}(0,1) & \text{otherwise} \end{cases}$$

ضریب $|\mathcal{I}_j(t)|$ نیز همان سیگما می باشد.

$$\sigma_i(t) = \alpha [x_j^{max}(t) - x_j^{min}(t)]$$

مراحل (روند اجرا) الگوریتم فرهنگی:

- ۱- تولید جمعیت اولیه و ارزیابی آنها
 - ۲- آماده سازی فضای اعتقادی
 - ۳- انتخاب اعضای جمعیت برای تاثیر بر فرهنگ
 - ۴- تغییر مولفه های فضای اعتقادی (فرهنگ) توسط اعضای منتخب
 - ۵- تولید پاسخ های جدید با تاثیر فرهنگ بر پاسخ های فعلی و ارزیابی آنها
 - ۶- بازگشت به مرحله سوم در صورت برآورده نشدن شرایط خاتمه
- که این روند بصورت الگوریتمی به شکل زیر میتواند مدل شود:

Begin

$t = 0;$

Initialize Population POP(t);

Initialize Belief Space BLF(t);

repeat

Evaluate Population POP(t);

Adjust(BLF(t), Accept(POP(t)));

Adjust(BLF(t));

Variation(POP(t) from POP(t-1));

until termination condition achieved

End

پیاده سازی اولیه الگوریتم فرهنگی:

الگوریتم پیاده سازی شده بر اساس استاندارد های بیان شده به پنج قسمت تقسیم بندی می شود که دسته های مختلف آن به شرح زیر است:

۱. تعریف مساله
۲. تعریف پارامترهای الگوریتم
۳. آماده سازی (Initialization)
۴. حلقه اصلی الگوریتم (پیاده سازی تکرار الگوریتم)
۵. نمایش نتایج

نیازمندی های پیاده سازی:

برای پیاده سازی نیاز به تابع sphere داریم. تابع sphere مجموع مربعات مولفه های یک بردار را محاسبه کرده و به دست می دهد.

$$f = f_{sph}(x) = \sum_{i=1}^n x_i^2$$

که مفروضاً x یک بردار باشد که حاوی مولفه های زیر باشد، نهایتاً تابع f مجموع توان دوم مولفه های آن بردار را محاسبه می کند.

کد مربوط به تابع sphere به شرح زیر است:

```
function z=Sphere(x)
```

```
    z=sum(x.^2);
```

```
end
```

برنامه اصلی را با سه دستور زیر آغاز می کنیم که بیانگر عملیات های زیر می باشد:

- دستور CLC : که صفحه ویرایشگر متن محیط متلب را پاک می کند.
- دستور clear : که حافظه موقت سیستم را از برنامه های پیشین اجرا شده توسط متلب پاک می کند.
- دستور close all : که تمامی توابع، خروجی ها، و نمودار های قبلی ایجاد شده توسط محیط متلب را می بندد.

قسمت اول – تعریف مساله :

در ابتدا تعریف تابع هزینه یا Cost Function را خواهیم داشت. که تابعی است از متغیر x که خروجی از نوع sphere(x) را به ما خواهد داد. تعداد متغیر های تصمیم را ۵ در نظر می گیریم. و آن را در یک بردار یک در پنج ذخیره می کنیم. البته تفاوت چندانی در انتخاب $1*5$ و یا $5*1$ بودن بردار وجود ندارد، چرا که در این الگوریتم محاسبات ماتریسی وجود ندارد (واضح است که در محاسبات ماتریسی همچون خوشه بندی و تشکیل dataset استفاده از بردارها با ابعاد عمودی بهینه تر می باشد). البته برای نمایش دادن داده ها استفاده از بردار افقی بهتر بنظر می رسد. اما در کل در این پیاده سازی نوع بعد بردار تصمیم تاثیری در فرایند محاسبات نخواهد داشت.

%% Problem Definition

CostFunction=@(x) Sphere(x); % Cost Function

nVar=5; % Number of Decision Variables (Unknown variables)

VarSize=[nVar]; % Decision Variables Matrix Size

VarMin=-10; % Decision Variables Lower Bound

VarMax= 10; % Decision Variables Upper Bound

پس اندازه ماتریسی که حاوی برداری که مولفه های آن متغیر های تصمیم الگوریتم خواهد بود ۵ عدد اتخاذ گردید. حد پایین و حد بالا بترتیب -۱۰ و +۱۰ انتخاب شدند.

قسمت دوم – تنظیمات مربوط به الگوریتم فرهنگی:

تعداد دفعات تکرار حلقه تکرار و تعداد اعضای جمعیت از جمله ملزومات تنظیمات اولیه یک الگوریتم تکاملی می باشد که در همین ابتدا مقداردهی می شوند.

پارامتر مهم بعدی نرخ افرادی است که انتخاب می شوند تا بر روی فضای اعتقادی تاثیر بگذارند. که این پارامتر با نام نرخ مقبولیت یا acceptance ratio خوانده می شود. در نهایت با ضرب کردن این نرخ در تعداد کل جمعیت تعداد جمعیت تاثیرگذار بر روی فرهنگ بدست می آید که در الگوریتم با نام nAccept پیاده سازی شده است. و در نهایت مقدار بدست آمده با استفاده از تابع round به یک مقدار صحیح گرد می شود.

%% Cultural Algorithm Settings

MaxIt=500; % Maximum Number of Iterations

nPop=50; % Population Size

pAccept=0.35; % Acceptance Ratio

nAccept=round(pAccept*nPop); % Number of Accepted Individuals

alpha=0.25;

beta=0.5;

که البته پیاده سازی فوق یک پیاده سازی ایستا (Static) می باشد. می توان این نرخ و طبعا این مقدار جمعیت را به شیوه پویا یا Dynamic نیز بدست آورد که در هر حلقه تکرار nAccept بدین صورت بدست آید:

for i=1 : it

 nAccept= round(nPop*gamma/it);

end

قسمت سوم – آماده سازی :

همانند همه الگوریتم های تکاملی هر پاسخ مفروض در فضای مساله دارای دو ویژگی بسیار مهم و حیاتی می باشد:

۱. موقعیت (Position): که موقعیت پاسخ مورد نظر را در فضای جست و جو نشان می دهد. (یک پاسخی را ارائه می کند).

۲. هزینه (Cost): هر پاسخ مقداری متناسب با تابع هدف را داراست که با استفاده از این ویژگی در مساله بیان خواهد شد.

در ابتدا یک ساختار خالی برای پذیرش مقادیر پاسخ ها تعریف می کنیم. واضحا هر کدام از این پاسخ ها یک موقعیت در فضا دارند.

سپس آماده سازی آرایه مربوط به اعضای جمعیت را انجام می دهیم. که این کار با استفاده از تابع `repmat()` که مخفف `repeat matrix` می باشد انجام می گیرد. لزوم استفاده از این تابع تکرار به اندازه تعداد اعضای جمعیت است. نتیجه نهایی تکرار شدن افقی یا زیر هم آرایه `empty_individual` می باشد. برای مقداردهی این مقادیر از یک حلقه تکرار استفاده می کنیم تا به ازای تمامی اعضای جمعیت این فرآیند را انجام دهد.

```
% Generate Initial Solutions
```

```
for i=1:nPop
```

```
    pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
```

```
    pop(i).Cost=CostFunction(pop(i).Position);W
```

```
end
```

موقعیت (`pop(i).Position`) عضو i ام جمعیت یک عدد تصادفی است در بازه تعریف شده جست و جو ، که با استفاده از تابع `uniform Random` این اعداد تصادفی تولید خواهند شد. و به همان ترتیب یک مقدار هزینه به تناسب با موقعیت آن تولید کرده و ذخیره می کنیم.

سپس بلافاصله مقادیر پاسخ ها را مرتب خواهیم کرد. برای مرتب سازی نیاز به مقدار هزینه تمامی اعضای جمعیت داریم. برای اینکار متغیر `pop.Cost` را بصورت یک ماتریس در میآوریم. که اینکار با `type Casting` و اضافه کردن علامت کروسه به عبارت در محیط متلب انجام می پذیرد.

```
% Sort Population
```

```
[~, SortOrder]=sort([pop.Cost]);
```

```
pop=pop(SortOrder);
```

خروجی تابع sort دو مقدار می باشد، مقدار اول که مقدار مرتب شده هزینه ها (مقادیر متناظر Cost برای هر پاسخ) می باشد. که نیازی به آن در این مقطع نداریم. پس مقدار خروجی اول را Null در نظر می گیریم. که این کار در محیط برنامه نویسی متلب با علامت تیلدا (~) انجام می پذیرد. مقدار خروجی دوم ترتیب قرارگیری اعضای جمعیت می باشد که sortOrder نامگذاری شده است.

pop=pop(sortOrder) نیز اعلام می دارد که اگر جمعیت ما همان جمعیت قبلی باشد با این تفاوت که طبق نحوه قرارگیری شان در آرایه مرتب بازچینی شده باشد، جمعیت حاصله یک جمعیت مرتب خواهد بود.

قسمت چهارم - بروزرسانی بهترین پاسخ یافت شده:

قسمت بعدی بروزرسانی بهترین پاسخ یافت شده می باشد:

```
% Update Best Solution Ever Found
BestSol=Culture.Situational;
```

یک آرایه نیز به منظور نگهداری بهترین پاسخ ها در هر تکرار در نظر می گیریم، در ابتدا مقادیر را با صفر پر می کنیم.

```
% Array to Hold Best Costs
BestCost=zeros(MaxIt,1);
```


قسمت پنجم – آماده سازی فرهنگ:

سپس نوبت به مهم ترین بخش آماده سازی یینی آماده سازی فرهنگ می رسد. یعنی پیاده سازی و بروزرسانی مولفه های هنجاری. اما در ابتدا لازم است که یک فرهنگ خالی ایجاد شود، سپس به بروزرسانی آن بپردازیم.

شایان ذکر است که در این نوع الگوریتم بخش فرهنگ بسیار مهم تر از پاسخ های فردی موجود در فضای مساله می باشد. به عبارتی پاسخ های مساله (در مساله مورد بررسی پاسخ های منفرد (individuals) فقط خودشان را توصیف می کنند. این در حالیست که فرهنگ مجموعه ای از پاسخ های منفرد (مجموعه ای از individual ها) را توصیف می کند. پس واضحاً یک مدل (در اینجا فرهنگ) می تواند بعدها به عنوان مبنایی برای استنتاج قرار گیرد. قسمت مربوط به آماده سازی فرهنگ به شکل زیر می باشد:

```
% Initialize Culture
Culture.Situational.Cost=inf;
Culture.Normative.Min=inf(VarSize);
Culture.Normative.Max=-inf(VarSize);
Culture.Normative.L=inf(VarSize);
Culture.Normative.U=inf(VarSize);
```

در ابتدا مقدار هزینه مولفه وضعی مقداری بسیار بزرگ جای گذاری می شود. سپس نوبت به جای گذاری مقادیر اولیه مولفه هنجاری می رسد.

دقت شود که هر پارامتری که به دنبال کمینه کردن آن می باشیم بایستی در ابتدا مقدار خام آن بی نهایت مثبت قرار گیرد. و متعاقباً هر پارامتری که به دنبال بیشینه کردن آن هستیم بایستی مقدار اولیه آن بی نهایت منفی جای گذاری شود. در قسمت بالا تنها پارامتری که بیشترین مقدار آن مد نظر است متغیر Culture.Normative.Max می باشد که تنها برای این متغیر مقدار اولیه منفی بی نهایت در نظر گرفته شده است.

علت وجود اسکریپت AdjustCulture.m در الگوریتم چیست؟

سوالی که پیش می آید این است که چگونه می توان جمعیت ای منتخب از فضای مساله را به یک فضای فرهنگی برده و مقادیر پاسخ های آنان را بروزرسانی کرد؟ در پاسخ به این سوال بایستی تابع ای تعریف شود که با ورودی و خروجی فرهنگ، مقادیر خروجی آن یک فرهنگ بوده، مقادیر ورودی آن فرهنگ و جمعیت انتخاب شده (جمعیتی که برای ورود به فرهنگ Accept شده اند) برای ورود به فضای فرهنگی باشد. که این امر در قطعه کد زیر قابل مشاهده می باشد:

```
function Culture=AdjustCulture(Culture,spop)
```

سپس نوبت به پیاده سازی شرط های مربوط به پیدا کردن مقادیر بهینه پاسخ ها در فضای فرهنگی می باشد. که این مهم با استفاده از الگوریتم زیر قابل دستیابی می باشد:

```
for i=1:n
    if spop(i).Cost<Culture.Situational.Cost
        Culture.Situational=spop(i);
    end

    for j=1:nVar
        if spop(i).Position(j)<Culture.Normative.Min(j) ...
            || spop(i).Cost<Culture.Normative.L(j)
            Culture.Normative.Min(j)=spop(i).Position(j);
            Culture.Normative.L(j)=spop(i).Cost;
        end
        if spop(i).Position(j)>Culture.Normative.Max(j) ...
            || spop(i).Cost<Culture.Normative.U(j)
            Culture.Normative.Max(j)=spop(i).Position(j);
            Culture.Normative.U(j)=spop(i).Cost;
        end
    end
end

Culture.Normative.Size=Culture.Normative.Max-Culture.Normative.Min;

end
```

به ازای تمامی عناصر جمعیت (متغیر n) و به ازای تمامی متغیرهای تصمیم (متغیر $nvar$) بایستی این تکرار انجام پذیرد. (همه اعضای جمعیت، همه متغیرها)

برای عضو i ام جمعیت و مولفه j ام آن داریم:

```
if spop(i).Position(j)<Culture.Normative.Min(j) ...
    || spop(i).Cost < Culture.Normative.L(j)
```

یا ثابت شود که پاسخ مورد نظر می تواند پاسخ بهینه تری را نسبت به پاسخ قبلی تولید کند (به نحوی میتوان بهینه تر بود). و یا پاسخ مورد بررسی رکوردشکنی کند و به عبارتی از مقدار هنجار پیش بینی شده هم بهینه تر و بهتر باشد. در این حالت خود پاسخ به عنوان پاسخ بهینه بروز رسانی می شود و علاوه بر این مقدار هزینه این پاسخ به عنوان هزینه بهینه آن هنجار در آن تکرار ذخیره می شود.

و به همین منوال برای مقادیر بزرگتری و بیشینه نیز کد مربوطه پیاده سازی می شود.

قسمت مربوط به بروز رسانی مولفه وضعی الگوریتم فرهنگی در ماژول `AdjustCulture.m` نیز به شرح زیر است:

```
for i=1:n
    if spop(i).Cost<Culture.Situational.Cost
        Culture.Situational=spop(i);
    end
```

اگر هزینه عضو i ام از جمعیت از مقدار رکورد مولفه وضعی کمتر باشد (رکورد مولفه وضعی را شکسته باشد) در آن صورت عضو i ام جمعیت به مولفه وضعی جدید جمعیت تبدیل می شود.

پس می توان بعد از مرتب کردن مقادیر جمعیت در ماژول اصلی `CA.m` مقدار فرهنگ را با استفاده از ماژول `AdjustCulture.m` بروز رسانی کرد.

```
% Adjust Culture
spop=pop(1:nAccept);
Culture=AdjustCulture(Culture,spop);
```

برای این کار در ابتدا آن دسته از جمعیت مورد بررسی مساله را که برای تاثیرگذاری در فرهنگ مقبول واقع شده اند را در متغیر دیگری بنام `spop` به معنای `selected population` ذخیره کرده و بروز رسانی فرهنگ جدید بر مبنای اجرای تابع `AdjustCulture` به ازای ورودی های جمعیت جدید انتخاب شده را انجام می دهیم.

قسمت مربوط به تاثیر گذاری فرهنگ بر روی جمعیت و تولید جمعیت جدید و بهبود یافته در حلقه اصلی الگوریتم فرهنگی دنبال می شود که به چند شیوه ای که قبلا اشاره شد (صرفا استفاده از هنجار، استفاده از مولفه وضعی برای تعیین جهت و مولفه هنجاری برای میزان حرکت)، می تواند پیاده سازی گردد.

حالت اول- صرفا استفاده از مولفه هنجاری

حالت دوم- صرفا استفاده از متغیر وضعی

حالت سوم - استفاده از مولفه وضعی برای تعیین جهت و مولفه هنجاری برای میزان حرکت

حالت چهارم - استفاده از مولفه هنجاری هم برای تعیین جهت و هم برای تعیین میزان گام

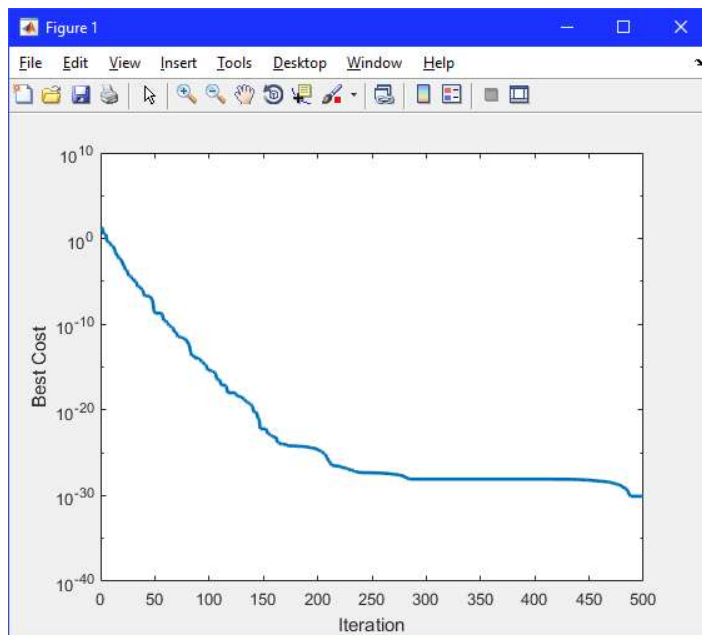
این پیاده سازی ها در پوشه "پیاده سازی" به انضمام فایل گزارش قابل دسترس می باشند.

اجرای الگوریتم فرهنگی در بستر

نرم افزار متلب

پس از طراحی الگوریتم و پیاده سازی آن در نرم افزار متلب با مقادیر ورودی زیر خروجی مطلوب به شرح زیر می باشد:

- جمعیت: ۵۰
- تعداد تکرار: ۵۰۰
- نرخ مقبولیت: ۰,۳۵
- نرخ آلفا (α): ۰,۲۵
- نرخ بتا (β): ۰,۵



نصیر ۸ خروجی الگوریتم فرهنگی پیاده سازی شده در نرم افزار متلب به ازای پارامترهای ورودی مذکور

پیاده سازی شبیه ساز تکامل دستورهای پخت غذا :

طرح مساله:

از آنجاییکه تمامی الگوریتم های تکاملی شهیر و مطرح در علم کامپیوتر برگرفته از اصول طبیعت و رفتار جمعی حیوانات و یا عناصر طبیعی و یا با استفاده از یک علت و یا معلول اجتماعی پایه گذاری می شوند، بر آن شدیم که الگوریتم فرهنگی پیاده سازی شده را در قالب یک الگوی اجتماعی/ فرهنگی جدید و قابل درک ارائه کنیم. از این جهت مدلی ارائه شد که روند تکامل فرهنگی را که از ذائقه ها و فرهنگ های مختلف نشات میگیرد را در قالب تکامل دستور پخت غذاهای مختلف پیاده سازی می کند.

بدین شکل که آشپزهای مختلف (عامل) هرکدام دستور پخت خاص خود را دارند. به مرور زمان و طی ارتباطات با دیگر آشپزها (با دستور پختها، ذائقه ها و گرایش غذایی متفاوت) تغییرات (در اینجا به تغییرات در نحوه پخت و یا آماده سازی غذاها تکامل گفته می شود). ای در نحوه تهیه دستور غذایی خود خواهند داد تا دستور غذایی واحد که مورد قبول تمامی جوامع آشپزی با تمامی تفاوت هایشان هستند برسیم.

از این نوع پیاده سازی می توان در یافتن علایق مشترک یا نحوه زندگی و یا حتی رئیس جمهور منتخب یک جامعه استفاده نمود.

شبیه ساز تکامل طرز تهیه خوراکی ها از سه قسمت اصلی تشکیل شده است:

- ۱- بخش مربوط به دستور تهیه (Recipe)
- ۲- بخش مربوط به عامل (Agent)
- ۳- بخش مربوط به تولید نسل جدید (Generation)

با اجرای برنامه کاربر قادر به تغییر تمامی پارامترهای دخیل در الگوریتم با استفاده از رابط کاربری (GUI) می باشد. و بعد از اجرا نتایج بصورت سلسه مراتبی در فایل متنی ذخیره شده تا اینکه بعد از خاتمه اجرای برنامه نیز نتایج در دسترس باشند. منظور از سلسه مراتبی بودن مراتب ذخیره نتایج این است که اطلاعات مربوط به تمامی عامل ها، گروه های اجتماعی و هر نسل در هرکدام از شبیه سازی های اجرا شده بصورت فایل متنی ذخیره میشود که بعدها قابل دسترس باشد.

کلاس Recipe.py دستور پخت ها را از وبگاه زیر دریافت می کند:

<http://mc6help.tripod.com/RecipeLibrary/RecipeLibrary.htm>

که این دستور پخت ها قبلا بارگذاری شده و در دایرکتوری پروژه در پوشه Recipes در دسترس هستند. لازم به ذکر است که دستور پخت ها در روند اجرای الگوریتم تاثیر داده نشده اند. دستور پخت ها بر اساس ذائقه های از پیش تعیین شده ی عامل ها دسته بندی می شوند.

پارامترهای مهم هر دستور پخت برای ما به شرح زیر است:

- ۱- عنوان (title)
- ۲- مدت زمان پخت یا آماده سازی
- ۳- مواد تشکیل دهنده
- ۴- Mutation history یا پیشینه نرخ جهش
- ۵- Evaluation score یا نتیجه ارزیابی

هر عامل در ابتدای کار با داشتن یکی از ذائقه های ماهی، گوشت و یا سبزیجات ایجاد می شود. در اولین نسل تعداد عامل های با ذائقه های متفاوت توسط کاربر در محیط کاربری تعیین می شود. هر عامل با توجه به ذائقه غذایی خود یکی از دستورات پخت مربوط به ذائقه خود را بصورت تصادفی انتخاب می کند. دستور پخت انتخاب شده تمام ویژگی های عامل را تعیین می کند. بدین شکل که مواد تشکیل دهنده دستور پخت انتخاب شده، مواد تشکیل دهنده محبوب آن عامل خواهند شد. بر اساس مدت زمان لازم برای پخت یا آماده سازی عامل ها ویژگی بنام سرعت تهیه پیدا میکنند که مقادیر "سریع"، "متوسط" و یا "کند" را به خود می گیرند.

بعد از طی کردن انتساب مقادیر و ویژگی های هر عامل، عامل غذای انتخاب شده خود را تهیه می کند. حتی خواهیم دید که مقادیر مواد اولیه را حذف اضافه و یا کم و زیاد کردن می کند. البته روشی که عامل دستور پخت خود را تغییر می دهد نیز به خودی خود بصورت تصادفی انتخاب می شود که این امر به نرخ MUTATION الگوریتم فرهنگی پیاده سازی شده بستگی دارد. هرگونه تغییرات در دستور پخت بوسیله عامل ممکن است، و هر تغییر به عنوان پیشینه تغییرات دستور پخت ذخیره و نگهداری می شود.

بطور دقیق تر فقط یکی از اعمال زیر و آن هم فقط یکبار (عامل فقط یک بار اجازه دخالت در دستور پخت خود را داراست) بوسیله عامل بر روی دستور پخت خود ممکن است:

- ۱- None کار خاصی انجام نشود
- ۲- اضافه کردن ماده تشکیل دهنده
- ۳- حذف ماده تشکیل دهنده
- ۴- تعویض یا تغییر ماده تشکیل دهنده

در پایان این مرحله، عامل دستور پخت خود را ارزیابی می کند. طبیعی است که غذای جدید ویژگی های غذای قبلی را نداشته باشد. و این موضوع در سیر تکامل عامل ها در الگوریتم های تکاملی امری طبیعی محسوب می شود. این عمل در مرحله ای از انجام الگوریتم بنام "مرحله ارتباطات" انجام می گیرد. تمامی این مراحل نیز در پیشینه فایل های متنی مربوط به هر عامل ذکر خواهد شد.

مراحل انجام الگوریتم فرهنگی بر روی دستورهای پخت:

بعد از طی مراحل اولیه، مرحله ارتباطات انجام می پذیرد که عامل با ارتباط با دیگر عامل ها دستور پخت خود را دچار تغییر می کند. تعداد دوستان و هم گروهی هایی که عامل می تواند با آنها به تبادل اطلاعات بپردازد نیز در رابط کاربری برنامه قابل تغییر می باشد. البته هر عامل می تواند فقط عضو یک گروه اجتماعی باشد. البته عامل ها بطور تصادفی در یکی از گروه های اجتماعی (برای بخش ارتباطات) عضو می شوند تا در یک گروه عامل های مختلفی با ذائقه های متفاوت (گوشت قرمز خواران ، ماهی خواران و یا سبزی خوارها) وجود داشته باشد.

مرحله ارتباطات بگونه ای فرایند انتخاب (selection) را شبیه سازی می کند. بطوریکه اعضای فرهنگ (در اینجا عامل های آشپزی) بر اساس شایستگی های خود ارزیابی و انتخاب می شوند. این شایستگی بر اساس امتیازی که برای هر عامل در نظر گرفته شده بوجود می آید. در فرایند ارتباطات در گروه های اجتماعی (social groups) براساس امتیازی که عامل ها به همدیگر اعطا می کنند محبوبیت و شایستگی عامل ها در گروه های اجتماعی مشخص می شود. این شایستگی تحت عنوان دوست داشتن (like) شبیه سازی شده است. بطوریکه یک امتیاز برای هر کدام از مشترکات مواد تشکیل دهنده ، دو امتیاز برای زمان تهیه یکسان و چهار امتیاز برای یکسان بودن دسته بندی خوراکی (گوشت قرمز ، ماهی و یا سبزیجات) در نظر گرفته می شود. در هر مرحله از اجرای selection این فرایند امتیازدهی توسط تمامی افراد گروه اجتماعی انجام می شود و مجموعه امتیازاتی که اعضای

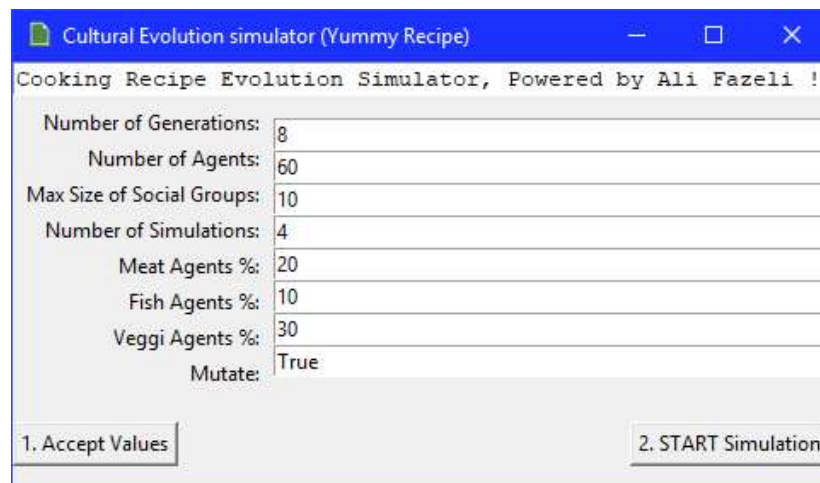
گروه به هریک از هم گروهی های خود می دهند امتیاز نهایی هر عامل و شایستگی آن را مشخص می کند. و در نهایت هر عامل با بیشترین میزان امتیاز در هر مرحله، به مرحله بعدی از افراد جمعیت منتقل می شود.

و این فرایند در نسل های بعدی جمعیت نیز تکرار شده و نسل های جدید بر اساس شایستگی بیشتر غربال می شوند. طبیعتاً دستور پخت عامل غربال شده در هر نسل و هر گروه اجتماعی نشان دهنده ذائقه محبوب و نظر جمعی آن گروه اجتماعی می باشد. البته تعداد دستور پخت ها (به تبع آن عامل ها) در طی زمان بشدت کاهش می یابد. بدین شکل که به محض غربال یکی از عامل ها در یکی از چند گروه اجتماعی (که بوسیله کاربر تعیین می شوند) شبیه سازی در آن مرحله به پایان می رسد و تنها یک عامل غربال شده از آن گروه به نسل بعدی انتقال می یابد و تمامی هم گروهی های آن حذف خواهد شد.

در آخر نتیجه نهایی یک دستور پخت خواهد بود که محبوب ترین و شایسته ترین دستور پخت موجود در تمامی نسل ها می باشد.

نحوه اجرای الگوریتم:

برای اجرای الگوریتم، نیاز به کتابخانه python نسخه ۲,۷ داریم. با در دست داشتن این کتابخانه، میتوان با اجرای اسکریپت Yummy_Recipe.py به محیط کاربری شبیه ساز وارد شویم.



تصویر ۹ محیط کاربری شبیه ساز

همانطور که در شکل بالا قابل مشاهده می باشد، میتوان شبیه سازی را مقدار دهی و اجرا کرد. در ابتدای کار مقادیر پیشفرض برای مقادیر در فیلدهای مربوطه از قبل نوشته شده اند. برای تغییر در هر کدام از فیلدها میتوان

مقدار مورد نظر را وارد نمود. در قسمت پایینی محیط امکان تغییر سهم درصدی عامل های هر بخش را تعیین کرد، بدین معنی که بطور مثال از مجموع دیتاست دستور پخت lamb.txt و beef.txt چند درصد دستور پخت ها در آزمایش شرکت داشته باشند. و این امر با توجه به ازدیاد دستور پخت ها انجام گردید. و در آخر میتوان تعیین کرد که نرخ جهش در انجام الگوریتم موثر باشد یا خیر.

لازم به ذکر است در صورت تعیین نشدن پارامتر ها و یا خالی گذاشتن آنها مقادیر پیشفرض طبق یک استثنای ورود (I/O Exception) به الگوریتم تزریق می شوند. بعد از وارد کردن مقادیر بر روی دکمه Accept Values کلیک کرده تا تمامی ورودی ها به شکل یک داده ساختار لیست دربیایند. سپس بر روی START Simulation کلیک می کنیم. مقادیر انتخاب شده تمامی عامل ها در پوشه Simulation و نتیجه نهایی محاسبات در فایل متنی بنام Stats.txt در مسیر اصلی اسکریپت ها ایجاد خواهند شد.

به عنوان مثال با اجرای الگوریتم با استفاده از مقادیر پیشفرض در خروجی Stats مقادیر زیر را داریم:

```

=====
||                               ||
||   Statistical Properties of this simulation run   ||
||                               ||
||=====||

Number of Agents           :    60
Maximum size of social groups :    10
Maximum number of Generations :     8
Number of individual CultEvo runs:    4

Values:
=====

Averages and measures of central location
.....

mean      :    7.00
median    :    7.00
mean_low  :    7.00
mean_high :    7.00
mean_grouped:    7.00
mode      :    7.00

Measures of spread
.....

Population standard deviation of data:    0.71
Population variance of data          :    0.50
Sample standard deviation of data     :    0.82
Sample variance of data               :    0.67

```

تصویر ۱۰_ مقادیر خروجی الگوریتم فرهنگی به ازای ورودی های پیشفرض

همانطور که قابل مشاهده است مقادیر ورودی الگوریتم در بخش اول دوباره نشان داده شده و مقادیر میانگین، میانه، مقادیر حداقل ای و حداکثری داده ها نیز نمایش داده شده اند.

در نهایت مقادیر انحراف معیار و واریانس داده های مورد بررسی (در اینجا دستور های پخت) نیز نمایش داده می شوند. اما نکته ای که بیش از هر چیز اهمیت دارد این است که در هر دوره از شبیه سازی در پوشه `Simulation` در دایرکتوری برنامه اطلاعات مربوط به تمامی عامل های درگیر بصورت فایل متنی ذخیره می شوند. در ابتدا به ازای هر عامل یک فایل متنی با نام شماره شناسایی آن عامل ایجاد شده، ذائقه آن ذکر شده و دستور پخت پیشنهادی آن عامل نیز ثبت می شود. سپس در هر شبیه سازی فایل متنی ای بنام `Gen_and_SocGrps.txt` ایجاد شده که اطلاعات جامعی از روند اجرای الگوریتم و تغییرات بوجود آمده بدست می دهد. در ابتدا تعداد عوامل موجود به تفکیک بخش مربوطه (گوشت قرمز، ماهی و یا سبزیجات) ذکر می شود. سپس تمامی گروه های اجتماعی (که در هنگام اجرای الگوریتم تعداد آنها توسط کاربر مقداردهی شده بودند) به همراه شماره شناسایی عامل های عضو هر گروه چاپ می شود.

بعد از آن به ازای تمامی نسل های مختلف، دستور پخت منتخب را بر اساس تعداد امتیاز های هریک (که هر کدام نیز جداگانه چاپ خواهند شد) به خروجی می روند.

نتیجه گیری و کارهای آینده:

از این بررسی میتوان در تحلیل نقاط مشترک یک جامعه دارای ویژگی های متعدد استفاده کرد. این ویژگی در پروژه مطالعه شده دستور پخت غذا های مختلف بوده است. اما یقیناً و همانطور که در مقدمه نیز اشاره گردید پتانسیل استفاده در الگوریتم های استخراج علایق و حتی بزرگتر در الگوریتم های تصمیم گیری جمعی نیز مورد استفاده قرار گیرد.

امید است در کارهای آینده به الگوریتم بخش `unit testing` نیز اضافه شود.

علاوه بر این میتوان الگوریتم را بصورت `abstract` بازتولید نمود تا بتوان در کارهای آینده با توجه به مساله مورد بررسی الگوریتم را شخصی سازی و به طبع آن بهینه سازی نمود.

منابع:

- ۱- An Introduction to Cultural Algorithms, Robert G. Reynolds
- ۲- Computational Intelligence, an introduction, Andries P. Engelbrecht. Cultural Algorithm section
- ۳- فیلم آموزشی وبگاه فرادرس، آشنایی با الگوریتم های فرهنگی، به نشانی
<http://faradars.org/courses/mvrca9107-cultural-algorithm-in-matlab-video-tutorial>
- ۴- اسلاید آموزشی تشابه الگوریتم ژنتیک و الگوریتم فرهنگی، به نشانی،
<https://www.slideshare.net/coddot/cultural-algorithm-genetic-algorithms-related-techniques>
- ۵-